Mathematical Programming 22 (1982) 304–315
North-Holland Publishing Company

POLYNOMIALLY BOUNDED ALGORITHMS FOR LOCATING p-CENTERS ON A TREE

R. CHANDRASEKARAN
The University of Texas at Dallas, Richardson, TX, U.S.A.

A. TAMIR
Tel-Aviv University, Tel-Aviv, Israel

Received 25 February 1979
Revised manuscript received 26 January 1981

We discuss several forms of the p-center location problems on an undirected tree network. Our approach is based on utilizing results for rigid circuit graphs to obtain polynomial algorithms for solving the model. Duality theory on perfect graphs is used to define and solve the dual location model.

Key words: Location Theory, Tree Networks, Rigid Circuit Graphs, Polynomial Algorithms.

1. Introduction

In this paper we present an efficient, polynomially bounded algorithm for determining p centers on an undirected tree network using the minimax criterion.

To formulate the problem precisely, we assume that an undirected tree $T = T(N, A)$, with N and A denoting the set of nodes and the set of arcs respectively, is given. T is embedded in the Euclidean plane, so that arcs are line segments whose endpoints are the nodes, and arcs intersect one another only at nodes. Moreover, each arc of T has a positive length. This embedding enables us to talk about points, not necessarily nodes, on the arcs. A is also used to denote the infinite set of points of T. For any two points $x, y \in A$, let $d(x, y)$ denote the distance between x and y measured along the arcs of T.

Two finite sets of points on T, S and D are specified. S is the set of possible locations for supply centers, while D represents the demand points. (Note that neither S nor D are assumed to be subsets of N, the set of original nodes of T.) Each demand point in D is associated with a positive number called the weight. Given a number of supply points, p, the objective is to find p locations in S for the p supply centers, such that the maximum of the weighted distances of the demand points to their respective nearest supply centers is minimized.

Following Hakimi [9], minimax location problems, discrete as well as continuous, on networks, have been studied quite extensively, with emphasis given to the algorithmic aspects. The main results appear in the following list of references: [8, 9, 10, 11, 12, 13, 17].

Focusing on a tree network $T = T(N, A)$, Handler [12], has suggested the categorization scheme $\{{}^N_A\}|\{{}^N_A\}|p$, where the first and second cells refer to the possible locations of facilities and demand points respectively, and the third cell indicates the number of supply centers to be established. This scheme identifies a variety of minimax facility location problems in tree networks. For example, $A|A|p$ refers to the problem of locating p centers, where each point on the tree is both a demand point and a potential location of a supply center.

Referring to the above categorization scheme, the recent work of Kariv and Hakimi [13] is the first to provide polynomially bounded algorithms for a general p. Prior to [13] efficient algorithms were given only for the special cases where p, the number of centers, was equal to 1 or 2, [4, 8, 9, 10, 11, 12, 17]. Kariv and Hakimi discuss only the models $N|N|p$ and $A|N|p$ and give an $O(n^2 \log n)$ procedure for solving these models ($n = |N|$.) Their work also contains several results on the complexity of minimax location problems on general graphs.

In this work we present a unified model for $N|N|p$, $A|N|p$ and $N|A|p$, and give a polynomially bounded algorithm for solving the weighted cases. (Weights are associated only with the models $N|N|p$ and $A|N|p$, i.e., when the set of demand points is finite.) A polynomial algorithm for $A|A|p$ has been recently developed and reported in [3].

Our approach was to relate the above location models to the general theory on perfect graphs, and in particular, to the class of rigid circuit graphs. In addition to providing polynomial algorithms, this approach enabled us to use the duality results on perfect graphs to define a dual to the above model. The dual is a problem of locating 'mutual obnoxious' facilities on the underlying tree network.

The organization of the paper is as follows. In Section 2, we present graph theoretic results on families of subtrees and neighborhood subtrees. These results are then used in Section 3 to develop an algorithm for the general weighted minimax location problem, described above, with general finite sets, D and S, of demand points and potential location points respectively. This algorithm has a worst case bound which is polynomial in $|N|$, $|D|$ and $|S|$. This bound will coincide with that of [13], when the algorithm is applied to the models $N|N|p$ and $A|N|p$. In Section 4, it is shown that the three cases $N|N|A$, $A|N|p$ and $N|A|p$ are special cases of the above general framework, with both sets S and D, containing at most $O(|N|^2)$ points. Section 5 focuses on computational aspects of the general algorithm. Section 6 is devoted to the dual location problem.

2. Intersection of trees and neighborhood trees

Considering a finite set of subtrees of T, $\{T_i\}$ $i = 1, \dots, m$, define the intersection graph, G, of $\{T_i\}$ as follows: G has m nodes, each corresponding to a different subtree in $\{T_i\}$. Two nodes are then connected by an (undirected) arc if and only if the two corresponding subtrees of T intersect.

Following [2], we note that the intersection graph G is a *rigid circuit graph*, i.e. each simple cycle of order greater than 3 contains a chord. Such graphs possess the following property due to Dirac [5] and reported also in [2].

Theorem 2.1. *Let G be a rigid circuit graph. Then G contains a node u such that u and all its neighboring nodes in G form a clique, i.e. the subgraph defined by u and its neighbors is a maximal complete subgraph of G.*

Nodes of G with the above property are called simplicial nodes. Also observe that the rigid circuit property is inherited. Namely, if a node and all its incident arcs are removed from a rigid circuit graph, then the remaining subgraph is also rigid circuit. In particular, this subgraph contains a simplicial node.

Next we prove that given a clique of the intersection graph G, the subtrees corresponding to the nodes of the clique have a common nonempty intersection, which is also a subtree of G.

Lemma 2.2. *Let T_1 and T_2 be subtrees of the tree T, with $T_1 \cap T_2 \neq \emptyset$. Then $T_1 \cap T_2$ and $T_1 \cup T_2$ are also subtrees of T.*

Lemma 2.3. *Let T_1, T_2 and T_3 be subtrees of the tree T, satisfying, $T_1 \cap T_2 \neq \emptyset$, $T_1 \cap T_3 \neq \emptyset$ and $T_2 \cap T_3 \neq \emptyset$. Then $T_1 \cap T_2 \cap T_3 \neq \emptyset$.*

Proof. Suppose that $(T_1 \cap T_3)$ does not intersect $(T_2 \cap T_3)$. Then since T_3 is connected, $S_3 = T_3 - ((T_1 \cap T_3) \cup (T_2 \cap T_3))$ is not empty. For $i = 1, 2$, let A_i be a point in $T_i \cap T_3$. Then there is a simple path P_1 on T_3 connecting A_1 and A_2. P_1 intersects S_3. A_1 and A_2 are also on the tree, $T_1 \cup T_2$, and therefore there exists a simple path P_2 on $T_1 \cup T_2$ connecting those two points. P_2 does not intersect S_3. Hence $P_1 \cup P_2$ contains a cycle-contradicting the tree property of T.

Theorem 2.4. *Let $\{T_i\}$, $i = 1, \dots, m$ be a set of subtrees of the tree T. If $T_i \cap T_j \neq \emptyset$ for all i, j, then $T_1 \cap T_2 \cap \dots \cap T_m$ is a nonempty tree in T.*

Proof. It is sufficient to prove that the intersection is nonempty. The proof is by induction on m. Assume $m \geq 4$. Consider the collection $R = \{T_1, T_2, \dots, T_{m-2}, \{T_{m-1} \cap T_m\}\}$. From Lemma 2.2 R consists of $m - 1$ nonempty subtrees, while Lemma 2.3 implies that the intersection of each pair in R is nonempty. By the induction hypothesis, the intersection of all of them, i.e. $\bigcap_{i=1}^{m} T_i$ is nonempty.

Next we define neighborhood trees and present several of their properties.

Definition 2.1. Let S be a finite set of points of T. Given a point p_i on T, and a number $r_i \geq 0$, the neighborhood tree of radius r_i, with center p_i, is the minimal subtree of T containing p_i and all points x in S with $d(p_i, x) \leq r_i$. This subtree is denoted by $T(p_i, r_i)$.

It is clear that $T(p_i, r_i)$ may consist of the center point p_i only. Furthermore, defining a tip to be a node of degree one, all the tips of $T(p_i, r_i)$, but possibly p_i, provided it is a tip, are points in S.

We now prove that if the intersection of a collection of neighborhood trees, each containing a point in S, is nonempty, then the intersection also contains a point in S.

Lemma 2.5. *Let $T(p_i, r_i)$ be a neighborhood tree in $T = T(N, A)$, which contains at least one point is S. Let x be a point in $T(p_i, r_i)$ and define $r = r_i - d(p_i, x)$. Then the neighborhood tree, $T(x, r)$, contained in $T(p_i, r_i)$, has at least one point of S.*

Proof. The result is obvious if $x = p_i$. Hence suppose $x \neq p_i$. Therefore, there exists a tip y of $T(p_i, r_i)$ such that y is in S and x is on the unique path connecting y with p_i. It is then clear that y is in $T(x, r)$.

Theorem 2.6. *For $i = 1, \ldots, m$ let $T(p_i, r_i)$ be a neighborhood tree in T, with radius r_i and center p_i. If $T(p_i, r_i)$, $i = 1, \ldots, m$, contains a point of S and $\bigcap_{i=1}^{m} T(p_i, r_i)$ is not empty, then $\bigcap_{i=1}^{m} T(p_i, r_i)$ contains a point of S.*

Proof. Let x be in $\bigcap_{i=1}^{m} T(p_i, r_i)$. For $i = 1, \ldots, m$ define $r'_i = r_i - d(x, p_i)$, and let j be such that $r'_j = \min_{1 \leq i \leq m} r'_i$. Then,

$$T(x, r'_j) \subseteq T(x, r'_i) \subseteq T(p_i, r_i) \quad \text{for all } i = 1, \ldots, m. \tag{2.1}$$

From Lemma 2.5 $T(x, r'_j)$ contains a point in S. (2.1) then implies that this point of S is also in $\bigcap_{i=1}^{m} T(p_i, r_i)$.

3. The general location model

Given a finite tree $T = T(N, A)$ with distances on the arcs, a finite set of points, $D \subseteq T$, corresponding to demand points is specified. Also, a finite set of points, $S \subseteq T$ at which supply centers can be located is identified. (Points of D or S are not necessarily original nodes of T. Also D and S may intersect.)

Further, there are weights associated with the demand points. Suppose that at most $p < |S|$ supply centers can be established. The objective is then to find the locations of the supply centers such that the maximum of the weighted distances of the demand points to their respective nearest supply center is minimized.

We introduce the following notation. Let $D = \{q_1, q_2, \dots, q_m\}$ be the demand points and let $S = \{s_1, s_2, \dots, s_k\}$ be the set of potential locations. $w_i > 0$, $i = 1, \dots, m$ will denote the weight associated with the demand point q_i.

The optimal maximum of the weighted distances of the demand points to their respective nearest supply points is equal to one of the following km numbers: $R = \{r_{ij} = w_i d(q_i, s_j)\}$, $i = 1, \dots, m$, $j = 1, \dots, k$. This latter observation suggests a procedure for the location model described above.

General scheme

(1) For each r in $R = \{r_{ij}\}$ find a subset $Y(r) \subseteq S$ of minimum cardinality such that

$$w_i \min_{s_j \in Y(r)} d(q_i, s_j) \le r \quad \text{for all } q_i \text{ in } D.$$

(2) Denoting by $p(r_{ij})$ the cardinality of the set $Y(r_{ij})$, the optimal location points for the supply centers is given by the set $Y(r_{ij})$ for which r_{ij} is the smallest among all values of r in R satisfying $p(r) \le p$ (p is the maximum number of supply centers that can be established).

We note in passing that if r^1, r^2 are in R and $r^1 < r^2$, then $p(r^1) \ge p(r^2)$. This monotonicity property enables one to reduce the computational effort required by the above general scheme. A further elaboration will be provided later.

Next we present a polynomially bounded algorithm, finding the subset $Y(r) \subseteq S$, of minimum cardinality, for an arbitrary $r \ge 0$, such that

$$w_i \min_{s_j \in Y(r)} d(q_i, s_j) \le r \quad \text{for all } q_i \text{ in } D.$$

Algorithm

Step 1. For each demand point q_i in D, find the neighborhood tree of radius $r_i = r/w_i$, $T(q_i, r_i)$, with respect to the set S of potential location points. If $T(q_i, r_i)$ contains no point of S, stop — the problem is infeasible.

Step 2. Generate the intersection graph, G, corresponding to the collection of neighborhood trees $\{T(q_i, r_i)\}$, $i = 1, \dots, m$.

Step 3. Find the minimum number of cliques covering all the nodes of G.

Step 4. For each clique found in Step 3, find a point of S in the intersection of the subset of neighborhood trees corresponding to the nodes of the clique.

$Y(r)$ consists of the points of S specified in Step 4 for the cliques in the minimum cover. The cardinality of $Y(r)$ is equal to the number of cliques in that cover.

To prove the validity of the algorithm, we first observe that given r the above problem is feasible if and only if each neighborhood tree, $T(q_i, r_i)$ contains at least one point of S. Equivalently, from Section 2, the problem is infeasible if and only if there exists a demand point q_i which is not a point in S, and $T(q_i, r_i) = \{q_i\}$.

Assuming feasibility, finding $Y(r)$ amounts to identifying the minimum number of points in S required to cover all the neighborhood trees, i.e. each tree $T(q_i, r_i)$ will contain at least one of these S points.

Given a supply point s_j, then the subset of neighborhood trees containing s_j corresponds to a complete subgraph in the intersection graph G. The results of Section 2 prove that we also have the reverse correspondence. More specifically, given a clique of G, Theorems 2.4–2.6 ensure that there exist a point of S, s_j, which is contained in all the neighborhood trees corresponding to the nodes of the clique. Moreover, the maximality of a clique as a complete subgraph shows that s_j is not contained in any tree which is not represented by a node of the clique.

The above discussion has validated the algorithm. We conclude this section by showing that the computational effort of the general scheme for solving the location problem is bounded by a polynomial in $m = |D|$, $k = |S|$, and $n = |N|$, the number of nodes in $T(N, A)$. The general scheme applies to the algorithm at most km times. In fact, since $p(r)$, defined in the general scheme, is a monotone piecewise function, the optimal r can be found using a binary search on R. Start by finding the median element r_1 in R and compute $p(r_1)$ by the algorithm. If $p(r_1) < p$, then the optimal r is greater than r_1, while if $p(r_1) \geq p$, r_1 is greater than or equal to the optimal r. In either case half of the elements in R can be omitted, and we proceed by searching the median r_2 of the remaining set R_1. Therefore, the algorithm is applied at most $O(\log_2(km))$ times until optimality is found. Also, note that the total effort of generating the entire sequence of medians $\{r_1, r_2, \ldots\}$ is $O(|R|) = O(km)$, since each time the cardinality of the remaining set is cut by half. (The median of a set can be found in linear time, [1].)

It is clear that Steps 1, 2 and 4 of the algorithm can be done in polynomial time. To find the minimum number of cliques covering all the nodes of the rigid circuit graph G, we can use the techniques of [7, 16]. There, it is shown how to find the minimum clique cover of a rigid circuit graph in $O(v + e)$ time, where v and e are the numbers of nodes and arcs of G, respectively.

Thus the general scheme is polynomially efficient.

4. Special cases

The following location problems on a tree $T = T(N, A)$ are discussed in the literature, [4, 9, 10, 11, 12, 13, 17].

(i) $N|N|p$. In this model the set of demand points, D, and the set of potential location points, S, are identical and equal to N, the set of the original nodes of $T(N, A)$. Given weights on the nodes, the objective is to locate at most p supply centers minimizing the weighted maximum of the distances to the respective nearest supply points.

(ii) $A|N|p$. The only difference between this model and $N|N|p$, is that here the supply points can be located anywhere on T.

(iii) $N|A|p$. In this problem supply centers can be located only at the nodes of T, i.e. $S = N$. The set of demand points consists of the whole continuum of points in T. There are no weights associated with the demand points, and the objective is to locate at most p supply points, minimizing the maximum of the distances to the respective nearest supply points.

(iv) $A|A|p$. The only difference between this model and $N|A|p$ is that the supply centers can be established anywhere along the continuum of points of T.

Next we show that the first three models described above are special cases of the general model of Section 3. Therefore, these models can be solved polynomially by the general scheme. The polynomiality of $N|N|p$ and $A|N|p$, for general p, on tree networks has only recently been established in [13]. A polynomial algorithm for $A|A|p$ is given in [3].

It is obvious that $N|N|p$ is a special case of the general model since we have $S = D = N$.

Turning to $A|N|p$, it is clear that $D = N$. Furthermore, the arguments given in [13, 14] demonstrate that it is sufficient to consider at most $|N|(|N|+1)/2$ potential locations for the supply points. This set of points consists of the N nodes and the $|N|(|N|-1)/2$ points obtained as follows. For each pair of nodes of T, q_i and q_j, with weights w_i and w_j respectively, consider the supply point x which is on the path connecting q_i and q_j, and $d(q_i, x) = w_j d(q_i, q_j)/(w_i + w_j)$. Thus, this model is also reduced to the general model with finite D and S.

Finally we turn to $N|A|p$. There we have $S = N$. The next lemma shows that although each point of T is a demand point, it is sufficient to assume that demand occurs only at the nodes and at the $|N|(|N|-1)/2$ midpoints of the paths connecting pairs of nodes.

Lemma 4.1. *Let $q_1, \dots, q_p$ be the nodes where the optimal p centers are located, for the model with $S = N$ and D consisting of the $|N|$ nodes of T and the $|N|(|N|-1)/2$ midpoints of the paths connecting pairs of nodes. Then the above nodes are also optimal locations for the model $N|A|p$.*

Proof. It is sufficient to show that for any setting of supply centers at nodes, the maximum of the distances of the demand points to their respective nearest supply points is attained for one of the above $|N|(|N|+1)/2$ demand points. Consider an arbitrary arc with nodes q_i and q_j, and let q_u be the nearest supply center to q_i. If q_u is also the nearest supply center to q_j, then q_u is the nearest supply point for every point x on the arc (q_i, q_j), with $d(x, q_u) \leq \max\{d(q_u, q_i),$

$d(q_u, q_j)\}$. Hence, suppose q_v is the supply center closest to q_j, where the entire arc (q_i, q_j) is on the path connecting q_u and q_v. Also, the midpoint of this path is on the arc (q_i, q_j). A simple calculation shows that the function $\min\{d(x, q_v), d(x, q_u)\}$ defined for x on the above path is maximized at its midpoint. This completes the proof.

5. Complexity and computational efficiency

To implement the general scheme, we assume that the sets S and D are input as follows. First, each node of T is appropriately labelled to indicate whether it belongs to S or D. Then, for each arc of T we have two separate lists for the subsets of S and D contained in the interior of that arc. These lists are sorted according to the distances of the points from one of the nodes on the arc, say, that node with the smallest index.

In the initial step of the general scheme we compute the set R of all distances between the demand points and the supply points. Generating this $m \times k$ distance matrix is done in total time of $O(m(n+k))$, since finding the distances from a given demand point to all points in S takes $O(|N|+|S|)$ time. We note that for the special cases considered in Section 4, this effort can be significantly reduced, since not all $m \times k$ distances are relevant.

Secondly we find the smallest element, r', in R such that

$$w_i \min_{s_j \in S} d(q_i, s_j) \le r'$$

for all q_i in D. Clearly, the elements of R smaller than r' can be omitted. r' is given by

$$r' = \max_{q_i \in D} \{w_i \cdot \min_{s_j \in S} d(q_i, s_j)\}.$$

Hence, to compute r', for each demand point we need its closest supply point. This can be done in $O(n + m \log k)$ as follows. For each node of T find its closest supply point. (An $O(n)$ time will suffice.) Then, for each interior demand point q_i an effort of $O(\log k)$ will suffice for finding the closest supply point located on the same arc. Comparing the distance of the latter with the distances to the closest supply points to the nodes of the arc gives the closest supply point to q_i.

Having computed r' we eliminate from R all the elements smaller than r', and turn to the algorithm. We have already noted above that the algorithm is applied at most $O(\log |R|)$ times in the general scheme. Given r in R satisfying $r \ge r'$ we next show how to implement the algorithm in $O(m(n+m+k))$. We start by finding the incidence matrix of the respective intersection graph G. Each node of G corresponds to a neighborhood tree of a demand point in D.

Given a demand point q_i we find all its neighbors in G as follows. Consider q_i as a node (possibly additional) of T. Find in $T(q_i, r/w_i)$ the furthest supply point from q_i on each simple path connecting q_i to a tip of T. Let $\bar{T}$ be the minimal subtree containing these supply points and q_i. It is easily seen that since $r \geq r'$, all demand points on $\bar{T}$ are neighbors of q_i. Furthermore, a demand point q_j which is not on $\bar{T}$ is a neighbor of q_i if and only if its respective neighborhood tree, $T(q_j, r_j)$, contains the supply point in $\bar{T}$ closest to q_j. This condition can be tested in $O(n + m)$ time for all demand points which are not in $\bar{T}$, provided that for each node of $\bar{T}$ its closest supply point in $\bar{T}$ has already been found. The latter can clearly be done in $O(n)$ time. Finally, finding $\bar{T}$ and all the demand points on $\bar{T}$ takes $O(m + n + k)$ time. Thus, we have demonstrated that all the neighbors of a given node in G are found in $O(m + n + k)$ time, and the entire incidence matrix is computed in $O(m(m + n + k))$ effort.

The next step of the algorithm is to find the minimum clique cover of G. This is done in $O(m^2)$ time, using the implementation of the algorithms of [7], as suggested in [16].

We observe that Step 4 of the algorithm is to be executed only for the optimal value of r in R. (For any iteration, but the last one, only $p(r)$, which is computed in Step 3 is needed for continuing the process.) Given the optimal r and the respective clique cover, evaluated in Step 3, we can execute Step 4 in $O(km)$ time.

Summarizing the above we obtain an $O(m(n + m + k)\log(km))$ bound for the general scheme.

Turning to the special cases considered in Section 4, we note that the above bound reduces to $O(n^2 \log n)$ for $N|N|p$, which is the same as the one obtained in [13].

For the model $A|N|p$ the implementation of the general scheme can be simplified. First, from [13] and the discussion in Section 4 it follows that it is not necessary to compute all the km distances between the demand and supply points. Instead, a set R consisting of $O(n^2)$ elements, (where each pair of nodes of T contributes an element), will suffice. The effort to generate this set R is $O(n^2)$. Also, an $O(n^2)$ test for the adjacency in G is available. Two nodes of G, say q_i and q_j, are adjacent if and only if the sum of the radii of their respective neighborhood trees, $r_i + r_j = r/w_i + r/w_j$, is not smaller than $d(q_i, q_j)$. Step 3 of the algorithm is done in $O(m^2) = O(n^2)$ time.

Finally, decomposing the nodes according to the clique cover and using the efficient algorithms for locating one center on a tree, [4, 10, 11, 13], we obtain the bound $O(n^2)$ for Step 4. Thus, the $A|N|p$ problem is also solved in $O(n^2 \log n)$ time, as in [13].

To our knowledge the $O(n^4 \log n)$ bound obtained for the $N|A|p$ model seems to be the first polynomial algorithm for this model.

6. Locating 'mutually obnoxious' facilities

The following location problem is highly related to the general location model described in Section 3.

Given the tree $T = T(N, A)$ and a finite set of points S in T we wish to place a fixed number of points, p, $p \leq |S|$, in S which are as far apart as possible from one another. As an application motivating this model we may consider the problem of locating ammunition depots, nuclear plants or emergency shelters against aerial attacks. We show that this problem of locating 'mutually obnoxious' facilities is equivalent, or dual to the problem of locating $p-1$ centers on T such that the maximum of the distances from the $k = |S|$ points of S to their respective nearest centers is minimized.

Let $\{\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_t\}$, $t = k(k-1)/2$, be the sorted sequence of distances on $T(N, A)$ between all distinct pairs of points in S. Assume that the above sequence contains only $r \leq t$ distinct values, which are then relabelled $W = \{\lambda_1 < \lambda_2 < \dots < \lambda_r\}$.

Lemma 6.1. *Let S_j, $j = 1, \dots, r$, be a subset of S with maximum cardinality such that the distances between distinct points in S_j is at least λ_j. Denote $N_j = |S_j|$.*

Let Q_j, $j = 1, \dots, r$ be a set of points in $T(N, A)$ of minimum cardinality such that the distances between points of S to their respective nearest points in Q_j is at most $\lambda_j/2$. Denote $P_j = |Q_j|$. Then $N_j = P_{j-1}$, $j = 1, \dots, r$. (We assume that $\lambda_0 = 0$).

Proof. From the previous sections we recall that P_{j-1} is the minimum number of cliques in the optimal clique cover of the nodes of the intersection graph G, corresponding to the $k = |S|$ neighborhood trees of radius $\lambda_{j-1}/2$.

To generate S_j, we first note that two points of S are in S_j if and only if the distance between them is at least λ_j. Since $\lambda_j > \lambda_{j-1}$, a distance between two points of S is at least λ_j if and only if it is greater than λ_{j-1}. Therefore, if G is the above intersection graph corresponding to the $k = |S|$ neighborhood trees of radius $\lambda_{j-1}/2$, then N_j is the cardinality of a maximum cardinality anticlique in G. (An anticlique is a maximal set of nodes in G no two are connected with an arc.)

Since G is a rigid circuit graph, we obtain that the cardinality of the largest anticlique is equal to the minimum number of cliques required to cover the nodes of G. Hence $N_j = P_{j-1}$, $j = 1, \dots, r$.

To introduce our duality result, suppose that $S = \{q_1, q_2, \dots, q_k\}$.

Theorem 6.2. *Given the tree $T = T(N, A)$, the finite subset $S \subseteq T$ and an integer $|S| \geq p > 1$, we have*

$$\max_{\substack{U \subseteq S \\ |U| = p}} \{\min\{d(q_i, q_j) \mid q_i, q_j \in U, q_i \neq q_j\}\} = 2 \min_{\substack{V \subseteq T \\ |V| = p-1}} \{\max_{q_i \in S} \{\min_{x \in V} d(q_i, x)\}\}.$$

Proof. Following [14], we observe that the sets V considered on the right hand side of the above relation can be assumed to be consisted only of the set of midpoints of the different paths connecting pairs of points in S. Hence, the right hand side is equal to λ_i, where λ_i is in W. Also the left hand side of the above relation is equal to λ_j, where λ_j is in W. We prove that $\lambda_j = \lambda_i$.

Using the notation of Lemma 6.1, λ_i is the smallest element in W such that $P_i \le p - 1$, and λ_j is the largest element in W with $N_j \ge p$, i.e. $m < i$ implies $P_m > p - 1$, and $m > j$ implies $N_m < p$.

Suppose that $j > i$. Then $i \le j - 1$ and therefore $P_i \ge P_{j-1}$. Applying Lemma 6.1 we obtain the contradiction

$$p \le N_j = P_{j-1} \le P_i \le p - 1.$$

Hence, $j \le i$. To see that also $j \ge i$, note that $N_i = P_{i-1} > p - 1$. Therefore, $N_i \ge p$. But j was defined as the largest element in W with $N_j \ge p$, which in turn yields $j \ge i$. We have thus shown that $i = j$ and therefore $\lambda_j = \lambda_i$. The proof is now complete.

A similar duality result dealing with the continuous problem of locating obnoxious facilities, i.e. S is assumed to be the whole continuum of points in $T(N, A)$, is presented in [17].

Finally, we turn to solving our problem of locating the obnoxious facilities. Referring to the set $W = \{\lambda_1 < \lambda_2 \dots < \lambda_r\}$, we have to find the largest λ_j such that $N_j \ge p$. The initial effort of finding the elements in W, or evaluating all the distances between the points in S is done in $O(k(n + k))$, where n is the number of nodes in $T(N, A)$ and $k = |S|$.

Next, given λ_i, finding N_i, is done by computing the largest anticlique on the corresponding intersection graph G. As mentioned in Sections 3–5, the techniques of [7, 16] can be utilized, yielding a bound of order $O(k^2)$ for finding N_i. The monotonicity of N_i in i, $i = 1, \dots, r$, enables us to solve the location problem by finding N_i for at most $O(\log k)$ values in W. Therefore, the total effort involved in finding the optimal locations is $O(kn + k^2 \log k)$.

7. Concluding comments

We have provided a unified approach and efficient algorithms for solving several classes of location problems on tree networks, by utilizing results on rigid circuit graphs. In particular, we applied the duality results on perfect graphs [6, 15], of which the rigid circuit graphs are only a subclass. In fact, the theory on perfect graphs allows us to generalize the model defined in Section 3, to reflect a variance in the demand among the points in D. We briefly define this extension, but omit the details since the solution procedure as well as the respective dual location model follow directly from the previous sections and [6, 15].

Suppose that for $i = 1, \dots, m$, demand point q_i is to be served by a_i centers. Then, consider the following covering problem, extending the problem of finding $Y(r)$ in Section 3. Given $r > 0$, find the minimum number, $p(r)$, of supply centers required such that at least a_i centers are set within a distance of r/w_i from q_i, $i = 1, \dots, m$. It is assumed that there is no bound on the number of centers that can be established at any supply point of S. To define the location model suppose that a total of p centers can be set. Then, the problem is to find the minimum r for which the respective covering problem satisfies $p(r) \leq p$.

References

[1] M. Blum, R.W. Floyd, V.R. Pratt, R.L. Rivest and R.E. Tarjan, "Time bounds for selection", *Journal of Computer and System Science* 7 (1973).

[2] P. Buneman, "A characterization of rigid circuit graphs", *Discrete Mathematics* 9 (1974) 205–212.

[3] R. Chandrasekaran and A. Daughety, "Problems of location on trees", Discussion paper No. 357, Center for Mathematical Studies in Economics and Management, Northwestern University (Evanston, 1978).

[4] P.M. Dearing and R.L. Francis, "A minimax location problem on a network," *Transportation Science* 8 (1974) 333–343.

[5] G.A. Dirac, "On rigid circuit graphs", *Abhandlung Mathematics Seminar* University of Hamburg 25 (1974) 71–76.

[6] D.R. Fulkerson, "Blocking and anti-blocking pairs of polyhedra", *Mathematical Programming* 1 (1971) 168–194.

[7] F. Gavril, "Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph", *SIAM Journal on Computing* 1 (1972) 180–187.

[8] A.J. Goldman, "Minimax location of facility in an undirected tree graph", *Transportation Science* 6 (1972) 407–418.

[9] S.L. Hakimi, "Optimum locations of switching centers and the absolute centers and medians of a graph", *Operations Research* 12 (1964) 450–459.

[10] S.L. Hakimi, E.F. Schmeichel and J.G. Pierce, "On p-centers in networks", *Transportation Science* 12 (1978) 1–15.

[11] G.Y. Handler, "Finding two-centers of a tree: the continuous case", *Transportation Science* 12 (1978) 93–106.

[12] G.Y. Handler, "Minimax location of a facility in an undirected tree graph", *Transportation Science* 7 (1973) 287–293.

[13] O. Kariv and S.L. Hakimi, "An algorithmic approach to network location problems, part 1: the p-centers", *SIAM Journal on Applied Mathematics* 37 (1979) 513–538.

[14] E. Minieka, "The m-center problem", *SIAM Review* 12 (1970) 138–139.

[15] M.W. Padberg, "Perfect zero–one matrices", *Mathematical Programming* 6 (1974) 180–196.

[16] D.J. Rose, R.E. Tarjan and G.S. Lueken, "Algorithmic aspects of vertex elimination on graphs", *SIAM Journal on Computing* 5 (1976) 266–283.

[17] D.R. Shier, "A min–max theorem for p-center problems on a tree", *Transportation Science* 11 (1977) 243–252.